Journal of Applied Logic 8 (2010) 422–440

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Complexity of hybrid logics over transitive frames ☆

Martin Mundhenk [a,*], Thomas Schneider [b], Thomas Schwentick [c], Volker Weber [c]

[a] *Institut für Informatik, Universität Jena, Germany*
[b] *Computer Science, University of Manchester, UK*
[c] *Fakultät für Informatik, Technische Universität Dortmund, Germany*

ARTICLE INFO

Article history:
Available online 12 August 2010

Keywords:
Hybrid logic
Satisfiability
Decidability
Complexity

ABSTRACT

This article examines the complexity of hybrid logics over transitive frames, transitive trees, and linear frames. We show that satisfiability over transitive frames for the hybrid language extended with the downarrow operator ↓ is NEXPTIME-complete. This is in contrast to undecidability over arbitrary frames (Areces et al. (1999) [2]). We also show that adding the @ operator or the past modality leads to undecidability over transitive frames. This is again in contrast to the case of transitive trees and linear frames, where we show these languages to be nonelementarily decidable. Furthermore, we establish 2EXPTIME and EXPTIME upper bounds for satisfiability over transitive frames and transitive trees, respectively, for the hybrid Until/Since language and complement them with an EXPTIME lower bound for the modal Until language.

1. Introduction

Hybrid languages are extensions of modal logic that allow for naming and accessing states of a model explicitly. This renders hybrid logic an adequate representation formalism for applications that require more expressivity than the basic modal or temporal language. Furthermore, reasoning systems are easier to devise for hybrid than for modal logic.

Hybrid Logic, as well as the foundations of temporal logic, goes back to Arthur Prior [35]. Since then, many—more or less powerful—languages have been studied. Here we briefly introduce the extensions that shall concern us in this article.

Nominals are special atomic formulae that name states of models. They allow, for instance, for an axiom expressing irreflexivity, which cannot be captured by modal formulae: $i \to \neg\Diamond i$.

The at operator @ can be used to directly jump to states named by nominals, independently of the accessibility relation. Hence, we can express that state i does not see itself, $@_i\neg\Diamond i$.

With the help of *the downarrow operator* ↓, it is possible to bind variables to states. Whenever $\downarrow x$ is encountered during the evaluation of a formula, the variable x is bound to the current state s. All occurrences of x in the scope of this ↓ are treated like nominals naming s. As an example, the formula $\downarrow x.\neg\Diamond\Diamond x$ reads as: Name the current state x and make sure that it is not possible to go from x to x in exactly two steps. This is an axiom for asymmetry, another property not expressible in modal logic.

Combined with the @ operator, ↓ leads to a very powerful language that can formulate many desirable properties and goes far beyond the scope of the simple nominal language. For a more impressive example, we consider the Until operator.

☆ A preliminary version of this article was presented at *Methods for Modalities 4* (2005). Supported in part by the grants DAAD-ARC D/08/08881, and BC-ARC 1323.

* Corresponding author.
E-mail addresses: martin.mundhenk@uni-jena.de (M. Mundhenk), schneider@cs.man.ac.uk (T. Schneider), thomas.schwentick@udo.edu (T. Schwentick).

1570-8683/$ – see front matter
doi:10.1016/j.jal.2010.08.004

The formula $\mathsf{U}(\varphi, \psi)$ reads as "there is a point in the future at which φ holds, and at all points between now and this point, ψ holds". This statement is again inexpressible in the basic modal language, but it can be said in the hybrid $\downarrow$-@ language.

$$\mathsf{U}(\varphi, \psi) \equiv \downarrow x.\Diamond\downarrow y.\varphi \wedge @_x\Box(\Diamond y \rightarrow \psi).$$

Besides more advanced temporal concepts such as "until" or "since", hybrid temporal languages can express other desirable temporal notions such as "now", "yesterday", "today", or "tomorrow". Moreover, with hybrid logic one can capture many temporally relevant frame properties (besides the above mentioned, antisymmetry, trichotomy, directedness, ...). For this reason, hybrid temporal languages are of great interest where basic temporal logic reaches its limits [9,6,4,17].

Transitive frames. Hybrid logic is interpreted over Kripke frames and models, as is modal logic. A frame consists of a set of states (points in time) and an accessibility relation R, where xRy says that y is reachable from x, for instance, y is in the future of x.

This article considers transitive frames because transitivity is a property that the relations of many different temporal applications have in common, even if they differ in other properties such as tree-likeness, trichotomy, irreflexivity, or asymmetry. Transitivity can be seen as the minimal requirement in many applications, for example temporal verification, which led to research on model-checking for hybrid logics [16].

There are more reasons why transitive frames are of interest, particularly in connection with computational complexity. In the special case of linear frames, nominals and @ can be defined using the conventional modal operator and its converse. Hence, the basic hybrid language is as expressive over linear frames as the basic modal language. The $\downarrow$ operator is useless even on transitive trees, a representation of branching time. Over transitive frames, in contrast, these hybrid operators *do* make a difference. In this case, there are properties that can be expressed in the hybrid, but not in the modal language (see the irreflexivity example above). For this reason, the class of transitive frames can be regarded as a restricted frame class that is still general enough to separate hybrid from modal languages in terms of expressive power.

Modal, hybrid, and first-order logics over transitive models have been studied recently in [3,18,41,25,26,24,13]. Although the complexity of hybrid (tense) logic has been extensively examined [7,19,2,3,17], there are highly expressive hybrid languages for whose satisfiability problems only results over arbitrary, but not over restricted, temporally relevant frame classes have been known. We examine the computational complexity of satisfiability for several hybrid logics over transitive frames and special cases thereof: transitive trees and linear frames.

Complexity of hybrid logics. We use complexity classes NP, PSPACE, EXPTIME, NEXPTIME, nEXPTIME, $n \geqslant 2$, and coRE as known from [34]. A problem is nonelementarily decidable if it is decidable but not contained in any nEXPTIME.

Obviously, reasoning tasks for richer logics require at least as many resources as those for simpler languages, often even significantly more. We focus on the most prominent reasoning task, satisfiability. The modal and temporal satisfiability problems over arbitrary as well as over transitive frames are PSPACE-complete [29,39]. If the "somewhere" modality E is added, satisfiability becomes EXPTIME-complete over arbitrary frames [38]. For many, more restricted, frame classes, modal and temporal satisfiability is NP-complete [29,33,37]. In contrast, the complexity spectrum of hybrid satisfiability reaches up to undecidability.

Many complexity results for hybrid languages have been established in [2,3]. It was proved in [2] that the hybrid language with nominals and @ has a PSPACE-complete satisfiability problem and that satisfiability for the hybrid tense language is EXPTIME-complete, even if @ or E are added. The same authors show that these problems have the same complexity (or drop to PSPACE-complete or NP-complete, respectively) if the class of frames is restricted to transitive frames (or transitive trees, or linear frames, respectively) [3].

Furthermore, [3] established EXPTIME-completeness of satisfiability for the hybrid Until/Since language. In contrast, PSPACE-completeness over linear frames is known from [17]. We are interested in the complexity over transitive frames and transitive trees, which, in terms of set inclusion, lie between the class of all frames and the class of linear frames. In general, the inclusion relation between frame classes does not imply any interreducibility of the corresponding satisfiability problems. Therefore, "our" open cases could theoretically have almost arbitrary complexity between coNP-hardness and coRE-containment.

Undecidability results for languages containing $\downarrow$ originate from [7,19]. The strongest such result, namely for the pure nominal-free fragment of the $\downarrow$ language, is given in [2]. A strong undecidability result with a unique state variable over the class of all frames can be also found in [31].

In [43], it was demonstrated that decidability of the $\downarrow$ language can be regained by certain restrictions on the frame classes. Transitivity could be another property under which the $\downarrow$ language can be "tamed", given that it has already been observed that over transitive trees and linear orders, the $\downarrow$ operator on its own is useless.

New road-map pages. This article considers the satisfiability problem of hybrid languages over transitive frames, transitive trees, and linear frames and establishes two groups of complexity results.

First, we examine the hybrid $\downarrow$ language. Our most important result is the "taming" of the $\downarrow$ language over transitive frames: satisfiability is NEXPTIME-complete. This high level of complexity is retained even over complete frames. We also show that enriching the language by the backward-looking modality P or the @ operator leads to undecidability in the case of transitive frames.

Table 1
An overview of complexity results for hybrid logics. Numbers in round parentheses refer to the corresponding theorem.

Hybrid lang.	Complexity over arbitrary frames	Complexity over transitive frames	Complexity over transitive trees	Complexity over linear orders
$\mathcal{HL}^{@}$	PSPACE [2]	PSPACE [3]	PSPACE [3]	NP [3]
$\mathcal{HL}_{\mathsf{F,P}}$	EXPTIME [2]	EXPTIME [3]	PSPACE [3]	NP [3]
$\mathcal{HL}^{\mathsf{E}}_{\mathsf{F,P}}$	EXPTIME [3]	EXPTIME [3]	PSPACE [3]	NP [3]
$\mathcal{HL}^{\mathsf{E}}_{\mathsf{U,S}}$	EXPTIME [3]	**in 2EXPTIME** (5.3), **EXPTIME-hard** (5.1)	**EXPTIME** (5.1), (5.4)	PSPACE-hard [36]
$\mathcal{HL}^{\downarrow}$	coRE [2]	**NEXPTIME** (3.1)	PSPACE [3]	NP [17]
$\mathcal{HL}^{\downarrow,@}$	coRE [2]	**coRE** (4.2)	**nonel.** (4.3)	**nonel.** (4.4)
$\mathcal{HL}^{\downarrow}_{\mathsf{F,P}}$	coRE [2]	**coRE** (4.2)	**nonel.** (4.3)	nonel. [17]
$\mathcal{HL}^{\downarrow,@}_{\mathsf{F,P}}$	coRE [2]	**coRE** (4.2)	**nonel.** (4.3)	nonel. [17]

The situation is different over transitive trees. Decidability, even for the richest $\downarrow$ language, is easy to see, but we will show a nonelementary lower bound for the languages that combine $\downarrow$ with P or @. For linear frames, the nonelementary lower bound for $\downarrow + \mathsf{P}$ is known; we will prove this bound for $\downarrow + @$.

We will also consider transitive frames and transitive trees for the hybrid language with Until, Since and E. We will establish EXPTIME-hardness for the modal language extended with Until only. This will be matched by an EXPTIME upper bound for the full language in the case of transitive trees. As for transitive frames, we will give a 2EXPTIME upper bound.

Table 1 gives an overview of the satisfiability problems considered in this article (marked bold) and shows our results in the context of previously known results. It makes use of the notation of hybrid languages introduced in Section 2. Complexity classes without addition stand for completeness results; "nonel." stands for nonelementarily decidable.

Legend. This article is organized as follows. In Section 2, we give all necessary definitions and notations of modal and hybrid logic. We present the decidability and undecidability results for the hybrid $\downarrow$ languages in Sections 3 and 4. The hybrid Until/Since language is examined in Section 5. Section 6 contains some concluding remarks.

Note. We mourn the loss of Volker Weber, who died suddenly and unexpectedly on the 7th of April 2009. He was 30 years old. Volker contributed an essential part to the present article which we finalized and submitted after his death.

2. Modal and hybrid logic

We define the basic concepts and notations of modal and hybrid logic that are relevant for our work; they are largely taken from [10,2,6].

Modal logic. Let PROP be a countable set of *propositional atoms*. The language $\mathcal{ML}$ of modal logic is the set of all formulae of the form

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Diamond\varphi,$$

where $p \in \text{PROP}$. We use the well-known abbreviations $\vee$, $\rightarrow$, $\leftrightarrow$, $\top$ ("true"), $\bot$ ("false"), and $\Box\varphi := \neg\Diamond\neg\varphi$. The set of all subformulae of φ is denoted by $\text{Sub}(\varphi)$.

The semantics is defined via *Kripke models*, which are triples $\mathcal{M} = (M, R, V)$, where M is a nonempty set of *states*, $R \subseteq M \times M$ is the *accessibility relation*, and $V : \text{PROP} \to \mathfrak{P}(M)$ is the *valuation function*. The structure $\mathcal{F} = (M, R)$ is called a *frame*. Given a model $\mathcal{M} = (M, R, V)$ and a state $m \in M$, the *satisfaction relation* is defined by

$$\begin{aligned}
\mathcal{M}, m \models p \quad &\text{iff} \quad m \in V(p), p \in \text{PROP},\\
\mathcal{M}, m \models \neg\varphi \quad &\text{iff} \quad \mathcal{M}, m \not\models \varphi,\\
\mathcal{M}, m \models \varphi \wedge \varphi' \quad &\text{iff} \quad \mathcal{M}, m \models \varphi \,\&\, \mathcal{M}, m \models \varphi',\\
\mathcal{M}, m \models \Diamond\varphi \quad &\text{iff} \quad \exists n \in M (mRn \,\&\, \mathcal{M}, n \models \varphi).
\end{aligned}$$

A formula φ is *satisfiable* if there exist a model $\mathcal{M} = (M, R, V)$ and a state $m \in M$, such that $\mathcal{M}, m \models \varphi$. If all states of $\mathcal{M}$ satisfy φ, we write $\mathcal{M} \models \varphi$ and say that φ is *globally satisfied* by $\mathcal{M}$.

Temporal logic. The language of temporal logic (tense logic) is the set of all formulae of the form

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \mathsf{F}\varphi \mid \mathsf{P}\varphi,$$

where $p \in \text{PROP}$. It is common practice to use the abbreviations $\mathsf{G}\varphi := \neg\mathsf{F}\neg\varphi$ and $\mathsf{H}\varphi := \neg\mathsf{P}\neg\varphi$. Satisfaction for F ("future") and P ("past") formulae is defined by

$$\begin{aligned}
\mathcal{M}, m \models \mathsf{F}\varphi \quad &\text{iff} \quad \exists n \in M (mRn \,\&\, \mathcal{M}, n \models \varphi),\\
\mathcal{M}, m \models \mathsf{P}\varphi \quad &\text{iff} \quad \exists n \in M (nRm \,\&\, \mathcal{M}, n \models \varphi).
\end{aligned}$$

$$
\begin{aligned}
&\mathcal{M}, g, m \models a && \text{iff} \quad m \in [V, g](a),\ a \in \text{ATOM},\\
&\mathcal{M}, g, m \models \neg\varphi && \text{iff} \quad \mathcal{M}, g, m \not\models \varphi,\\
&\mathcal{M}, g, m \models \varphi \wedge \varphi' && \text{iff} \quad \mathcal{M}, g, m \models \varphi \,\&\, \mathcal{M}, g, m \models \varphi',\\
&\mathcal{M}, g, m \models \Diamond\varphi && \text{iff} \quad \exists n \in M (mRn \,\&\, \mathcal{M}, g, n \models \varphi),\\
&\mathcal{M}, g, m \models @_t\varphi && \text{iff} \quad \exists n \in M\big(\mathcal{M}, g, n \models \varphi \,\&\, [V, g](t) = \{n\}\big),\\
&\mathcal{M}, g, m \models \downarrow x.\varphi && \text{iff} \quad \mathcal{M}, g^x_m, m \models \varphi.
\end{aligned}
$$

Fig. 1. The satisfaction relation for hybrid formulae.

Whenever we want to speak not only of states accessible from the current state, but also of states "between" the current and some accessible state, we can make use of the binary operators U ("until") and S ("since"), for which satisfaction is defined by

$$
\begin{aligned}
&\mathcal{M}, m \models \mathsf{U}(\varphi, \psi) \quad \text{iff} \quad \exists n\big(mRn \,\&\, \mathcal{M}, n \models \varphi \,\&\, \forall s(mRsRn \Rightarrow \mathcal{M}, s \models \psi)\big),\\
&\mathcal{M}, m \models \mathsf{S}(\varphi, \psi) \quad \text{iff} \quad \exists n\big(nRm \,\&\, \mathcal{M}, n \models \varphi \,\&\, \forall s(nRsRm \Rightarrow \mathcal{M}, s \models \psi)\big).
\end{aligned}
$$

The U/S language is strictly stronger than the basic temporal language in the sense that F and P can be expressed via U and S (e.g. $\mathsf{F}\varphi = \mathsf{U}(\varphi, \top)$), but not vice versa.

In [3], a variant of the U/S operators, U^+ and S^+, is introduced. Satisfaction for U^+ (analogously for S^+) is defined by

$$
\begin{aligned}
&\mathcal{M}, m \models \mathsf{U}^+(\varphi, \psi)\\
&\quad \text{iff} \quad \exists n \in M\big(mRn \,\&\, \mathcal{M}, n \models \varphi \,\&\, \forall s \in M\big(mR^+sR^+n \Rightarrow \mathcal{M}, s \models \psi\big)\big),
\end{aligned}
$$

where R^+ is the transitive closure of R. By means of these operators, they "simulated" transitive frames syntactically [3].

We go a step further and define another modification, U^{++} and S^{++}, with the satisfaction relation

$$
\begin{aligned}
&\mathcal{M}, m \models \mathsf{U}^{++}(\varphi, \psi)\\
&\quad \text{iff} \quad \exists n \in M\big(mR^+n \,\&\, \mathcal{M}, n \models \varphi \,\&\, \forall s \in M\big(mR^+sR^+n \Rightarrow \mathcal{M}, s \models \psi\big)\big),
\end{aligned}
$$

and analogously for S^{++}. The resulting temporal language is an even closer simulation of transitivity, as we will see in Section 5.

Hybrid logic. As indicated in the previous section, there are several extensions of the modal language that allow for explicit references to states and which therefore are called hybrid. We introduce the hybrid languages that will be of interest to us in this article. The definitions and notations are taken from [2,3].

Let NOM be a countable set of *nominals*, SVAR a countable set of *state variables*, and $\text{ATOM} = \text{PROP} \cup \text{NOM} \cup \text{SVAR}$. As usual, we denote propositional atoms by $p, q, \ldots$, nominals by $i, j, \ldots$, and state variables by $x, y, \ldots$. The *full hybrid language* $\mathcal{HL}^{\downarrow,@}$ is the set of all formulae of the form

$$\varphi ::= a \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Diamond\varphi \mid @_t\varphi \mid \downarrow x.\varphi,$$

where $a \in \text{ATOM}$, $t \in \text{NOM} \cup \text{SVAR}$, and $x \in \text{SVAR}$.

A hybrid formula is called *pure* if it contains no propositional atoms; *nominal-free* if it contains no nominals; and a *sentence* if it contains no free state variables. (*Free* and *bound* are defined as usual; the only binding operator here is $\downarrow$.)

A *hybrid model* is a Kripke model with the valuation function V extended to $\text{PROP} \cup \text{NOM}$, where for all $i \in \text{NOM}$, it holds that $|V(i)| = 1$. Whenever it is clear from the context, we will omit the word "hybrid" when referring to models. In order to evaluate $\downarrow$-formulae, an *assignment* $g : \text{SVAR} \to M$ *for* $\mathcal{M}$ is necessary. Given an assignment g, a state variable x and a state m, *an x-variant* g^x_m *of* g is defined by

$$g^x_m(x') = \begin{cases} m & \text{if } x' = x,\\ g(x') & \text{otherwise.} \end{cases}$$

For any atom a, let

$$[V, g](a) = \begin{cases} \{g(a)\} & \text{if } a \in \text{SVAR},\\ V(a) & \text{otherwise.} \end{cases}$$

The satisfaction relation for hybrid formulae is given in Fig. 1. For sentences φ we may also write $\mathcal{M}, m \models \varphi$ to denote that $\mathcal{M}, g, m \models \varphi$ for some g for $\mathcal{M}$. The latter condition is equivalent to "$\mathcal{M}, g, m \models \varphi$ for *every* g for $\mathcal{M}$". A formula is *satisfiable* if there exist a model $\mathcal{M} = (M, R, V)$, an assignment g for $\mathcal{M}$, and a state $m \in M$ such that $\mathcal{M}, g, m \models \varphi$.

We sometimes use the "somewhere" modality E having the interpretation

$$\mathcal{M}, g, m \models \mathsf{E}\varphi \quad \text{iff} \quad \exists n \in M(\mathcal{M}, g, n \models \varphi).$$

In this case, $@_t\varphi$ can be expressed by $\mathsf{E}(t \wedge \varphi)$.

First-order logic. Modal and hybrid logic can be embedded into fragments of first-order logic. We will always use the standard notation of first-order logic.

We will make use of certain fragments of first-order logic and denote them in the style of [11]. Let S be a quantifier sequence, for instance $\exists\forall\forall$ or "all" which stands for arbitrary sequences. Let $p_1, \ldots, p_n, f_1, \ldots, f_m$ be natural numbers or ω, for $n, m \in \mathbb{N}$. Let $(p_1, \ldots, p_n)$ and $(f_1, \ldots, f_m)$ denote two sets that consist of exactly p_i i-ary predicate symbols for $i = 1, \ldots, n$ and f_j j-ary function symbols for $i = j, \ldots, m$, respectively. The notation $[S, (p_1, \ldots, p_n), (f_1, \ldots, f_m)]$ stands for the set of all first-order formulae that use quantifiers only in prefixes S and which contain relations and function symbols taken only from the sets $(p_1, \ldots, p_n)$ and $(f_1, \ldots, f_m)$. We denote the satisfiability problem for such a fragment by $[S, (p_1, \ldots, p_n), (f_1, \ldots, f_m)]$-SAT.

The first-order fragments that we are interested in allow no function symbols and, in one case, an additional binary equality predicate. We omit the $f_1, \ldots, f_m$ and denote these fragments by $[\text{all}, (p_1, 1)]$ and $[\text{all}, (\omega), 0]_=$. The latter is called the *Monadic Class with equality* $\mathsf{MC}_=$ [11]. We use $[\text{all}, (p_1, 1)]$-trans-SAT to denote the satisfiability problem of $[\text{all}, (p_1, 1)]$ where the binary predicate symbol is required to be interpreted by a transitive relation.

The *Standard Translation* ST (see e.g. [10]) embeds hybrid logic into first-order logic and consists of two functions ST_x and ST_y, defined recursively. Since ST_y is obtained from ST_x by exchanging x and y, we only give ST_x here.

$$\begin{aligned}
&\mathrm{ST}_x(p) = P(x), && \mathrm{ST}_x(\Diamond\varphi) = \exists y\big(xRy \wedge \mathrm{ST}_y(\varphi)\big),\\
&\mathrm{ST}_x(t) = t = x, && \mathrm{ST}_x(@_t\varphi) = \exists y\big(y = t \wedge \mathrm{ST}_y(\varphi)\big),\\
&\mathrm{ST}_x(\neg\varphi) = \neg\mathrm{ST}_x(\varphi), && \mathrm{ST}_x(\downarrow v.\varphi) = \exists v\big(x = v \wedge \mathrm{ST}_x(\varphi)\big),\\
&\mathrm{ST}_x(\varphi \wedge \psi) = \mathrm{ST}_x(\varphi) \wedge \mathrm{ST}_x(\psi), && \mathrm{ST}_x(\mathsf{E}\varphi) = \exists y\big(\mathrm{ST}_y(\varphi)\big),
\end{aligned}$$

where $p \in \mathrm{PROP}$, $t \in \mathrm{NOM} \cup \mathrm{SVAR}$, and $v \in \mathrm{SVAR}$. The Standard Translation shows that hybrid logic can be seen as an undecidable fragment of first-order logic that has several nice properties [43].

Properties of models and frames. Let $\mathcal{M} = (M, R, V)$ be a (Kripke or hybrid) model with the underlying frame $\mathcal{F} = (M, R)$. For any subset $M' \subseteq M$, we write $R\restriction_{M'}$ and $V\restriction_{M'}$ for the restrictions of R and V to M'. We will refer to *transitive frames* or *linear frames* whenever we mean frames whose accessibility relation is transitive or a linear order, respectively. A frame $\mathcal{F}$ is a *tree* if it is acyclic and connected and every point has at most one R-predecessor. A *transitive tree* is any (M, R^+) where (M, R) is a tree. A *transitive model* is a model whose underlying frame is transitive. A *linear order* is an irreflexive, transitive, and trichotomous relation, where trichotomy is defined by $(\forall xy(xRy \text{ or } x = y \text{ or } y\ Rx))$.

Satisfiability problems. We consider fragments of the full hybrid language $\mathcal{HL}^{\downarrow,@}$ by leaving out hybrid operators. To denote these fragments, we omit the corresponding superscript of $\mathcal{HL}$. For hybrid tense languages, we add the suitable temporal operator(s) as subscript(s) to $\mathcal{HL}$. Analogously, when equipping the modal language with additional operators, we add them as sub- or superscripts to $\mathcal{ML}$.

For any hybrid language $\mathcal{HL}^x_y$, the *satisfiability problem* $\mathcal{HL}^x_y$-SAT is defined as follows: Given a formula $\varphi \in \mathcal{HL}^x_y$, does there exist a hybrid model $\mathcal{M}$, an assignment g for $\mathcal{M}$, and a state $m \in M$ such that $\mathcal{M}, g, m \models \varphi$? If $\downarrow$ is not in the considered language, the assignment g can be omitted as on page 425. If we only ask for *transitive models* (or *transitive trees* or *linear models*, respectively) satisfying φ, we write $\mathcal{HL}^x_y$-trans-SAT (or $\mathcal{HL}^x_y$-tt-SAT, or $\mathcal{HL}^x_y$-lin-SAT, respectively). For instance, the satisfiability problem over transitive frames for the hybrid temporal $\downarrow$ language is denoted by $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-trans-SAT.

3. Deciding $\mathcal{HL}^{\downarrow}$ over transitive frames

Areces, Blackburn, and Marx [2] proved that the downarrow operator $\downarrow$ turns the satisfiability problem for hybrid logics undecidable in general, even if no interaction with @ or P is allowed.

We prove that decidability is regained if frames are required to be transitive.

Theorem 3.1. *The satisfiability problem for $\mathcal{HL}^{\downarrow}$ over transitive frames is complete for NEXPTIME.*

As preparations for the proof, we have a look at transitive models for $\mathcal{HL}^{\downarrow}$. Obviously, $\mathcal{HL}^{\downarrow}$ has no finite model property. For example, the following sentence requires a model containing an infinite chain of states labeled p.

$$p \wedge \Diamond p \wedge \Box\Diamond p \wedge \Box\downarrow x.\neg\Diamond x.$$

Over transitive frames, $\mathcal{HL}^{\downarrow}$ does not have the tree model property either. However, we can approximate transitive tree models by unifying cycles: due to transitivity, all states in a cycle are pairwise connected, i.e., the subframe consisting of

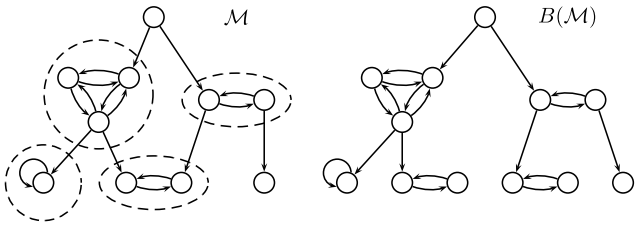

Fig. 2. A transitive model $\mathcal{M}$ and its block tree $B(\mathcal{M})$. The maximal complete subframes of $\mathcal{M}$ are marked by the dashed circles. Some edges caused by transitivity are left out for simplicity.

these states is complete. Therefore, we can consider a model as consisting of maximal complete subframes and single states, all of which are connected in a transitive but acyclic fashion.

For every transitive model $\mathcal{M} = (M, R, V)$, we define its *block tree* $B(\mathcal{M}) = (M', R', V')$ as the structure obtained as follows. First, we replace each maximal complete subframe of $\mathcal{M}$—clique, for short—with a single vertex. Second, we unravel the resulting structure into a (potentially infinite) transitive tree T. Then we replace each vertex of T with (a copy of) the clique of $\mathcal{M}$ from which it is derived (Fig. 2).

We refer to the cliques of a block tree as *nodes*. For a state s, we denote its node by u_s. We say that a node v *is below* a node u if the states of v are reachable from the states in u (but not vice versa) and a node v is a *child* of a node u, if v is below u but there is no node w below u and above v. We usually omit the word "block" from the terms "tree", "subtree" and "leaf".

Nominals are not preserved when unraveling a model: if $V(i) = \{s\}$ for a nominal i and a state $s \in M$, we define $V'(i)$ to be the set of states from M' that are copies of s via the unraveling, treating i as a propositional atom in $B(\mathcal{M})$. Now it is easy to see that the transformation from $\mathcal{M}$ to $B(\mathcal{M})$ preserves satisfaction of $\mathcal{HL}^{\downarrow}$-formulae: the relation associating each state of $\mathcal{M}$ with every copy in $B(\mathcal{M})$ is a quasi-injective bisimulation, i.e., a bisimulation with the additional requirement that states labelled with the nominal i in $\mathcal{M}$ and with the atomic proposition i in $B(\mathcal{M})$ are related [8].

Lemma 3.2. *For every transitive model $\mathcal{M}$, every state s of $\mathcal{M}$, every copy s' of s in $B(\mathcal{M})$, and every $\mathcal{HL}^{\downarrow}$-sentence φ:*

$$\mathcal{M}, s \models \varphi \quad \Longleftrightarrow \quad B(\mathcal{M}), s' \models \varphi.$$

For an explanation of the omitted assignment g, see page 425. If, for some block tree $\mathcal{B}$ and some state s of $\mathcal{B}$, it holds that $\mathcal{B}, s \models \varphi$, we refer to $\mathcal{B}$ as a block tree model for φ.

Before we show how to use block trees to decide $\mathcal{HL}^{\downarrow}$ over transitive frames, we will show that the size of cliques can be bounded, considering the satisfiability problem of $\mathcal{HL}^{\downarrow}$ over complete frames.

3.1. $\mathcal{HL}^{\downarrow}$ over complete frames

We will prove an exponential-size model property of $\mathcal{HL}^{\downarrow}$ over complete frames and use it to show NEXPTIME-completeness of $\mathcal{HL}^{\downarrow}$-trans-SAT. The upper bound will easily carry over to $\mathcal{HL}^{\downarrow,@}_{\mathsf{F,P}}$, too.

In complete frames, states can be told apart only if they are labeled differently by propositions or are assigned to different names. The number of different labelings is exponentially bounded in the size of the formula, and the number of states we can distinguish by different names is bounded by the number of different state variables and nominals used in the formula.

Theorem 3.3. *$\mathcal{HL}^{\downarrow}$ over complete frames has the exponential-size model property and its satisfiability problem is complete for NEXPTIME.*

Proof. We observe that $\mathcal{HL}^{\downarrow}$ over complete frames is equivalent to $\mathsf{MC}_{=}$, which is NEXPTIME-complete [11, Section 6.2.1]. Moreover, every hybrid model with a complete frame can be viewed as a relational structure for $\mathsf{MC}_{=}$, and vice versa.

Claim. *The satisfiability problems for $\mathcal{HL}^{\downarrow}$ over complete frames and for $\mathsf{MC}_{=}$ are polynomial-time equivalent.*

Proof. The accessibility relation in complete frames can be ignored when going from $\mathcal{HL}^{\downarrow}$ to $\mathsf{MC}_{=}$. The reduction to $\mathsf{MC}_{=}$ is the Standard Translation ST as defined in Section 2 with the rule for the diamond operator replaced by $\mathrm{ST}_x(\Diamond\alpha) = \exists y(\mathrm{ST}_y(\alpha))$.

For the converse direction, we define a reduction HT as follows.

$$\mathrm{HT}(\bot) = \bot, \qquad \mathrm{HT}\big(P(x)\big) = \Diamond(x \wedge p),$$
$$\mathrm{HT}(\varphi \wedge \psi) = \mathrm{HT}(\varphi) \wedge \mathrm{HT}(\psi), \qquad \mathrm{HT}(\exists x.\varphi) = \Diamond\big(\downarrow x.\mathrm{HT}(\varphi)\big),$$

$$\mathrm{HT}(\neg\varphi) = \neg\mathrm{HT}(\varphi), \qquad \mathrm{HT}(x = y) = \Diamond(x \wedge y).$$

Both reductions can be computed in polynomial time. □

We can therefore transfer complexity results and model properties for $\mathsf{MC}_=$ [11] to $\mathcal{HL}^{\downarrow}$ over complete frames. □

The lower bound can be transferred directly to the case of transitive frames.

Corollary 3.4. *$\mathcal{HL}^{\downarrow}$-trans-SAT is NEXPTIME-hard.*

Proof. We reduce $\mathsf{MC}_=$ to $\mathcal{HL}^{\downarrow}$-trans-SAT via the function that maps a given $\mathsf{MC}_=$-formula φ to the $\mathcal{HL}^{\downarrow}$-formula $(\downarrow x.\Box\Diamond x) \wedge \mathrm{HT}(\varphi)$, where the first disjunct forces the transitive subframe generated by the initial state to be complete. □

3.2. On transitive frames for $\mathcal{HL}^{\downarrow}$

So far we know that, for every $\mathcal{HL}^{\downarrow}$-formula φ satisfiable over transitive frames, we can consider its block tree whose cliques are of size at most exponential in $|\varphi|$.

The algorithm for testing $\mathcal{HL}^{\downarrow}$-satisfiability will guess such a block tree model and verify that it is correct. Satisfying models can still be infinite, but we will show that they have finite representations, similarly to tableau branches.

To this end, we define the φ-type of a state. It captures the information needed about its subtrees in order to evaluate any subformula of a given formula φ at this state. Here, $\psi[\mathit{free}/\bot]$ is the formula obtained from ψ by replacing every free variable by $\bot$.

Definition 3.5. Let φ be an $\mathcal{HL}^{\downarrow}$-sentence and $\mathcal{B} = (M, R, V)$ a block tree model. The *φ-type* of a state $s \in M$ is the set of all sentences from $\{\psi[\mathit{free}/\bot] \mid \Diamond\psi \in \mathrm{Sub}(\varphi)\}$ that hold at some state in the subtree rooted at s.

Note that states in the same clique have the same φ-type. Therefore, we can speak of the φ-type of a node. The type of a node is always a superset of the types of its children.

When evaluating a subformula of an $\mathcal{HL}^{\downarrow}$-sentence φ at some state s of a block tree, all we need to know about states strictly below u_s are the φ-types of the children of u_s. I.e., we can replace subtrees below u_s by subtrees of the same φ-type. In the following lemma, for a block tree $\mathcal{B}$ and two states s_1, s_2, $\mathcal{B}[u_{s_1}/u_{s_2}]$ denotes the block tree resulting from $\mathcal{B}$ by replacing the subtree rooted at u_{s_1} by the subtree rooted at u_{s_2}. The result of this substitution is again a block tree that preserves satisfaction.

Lemma 3.6. *Let φ be an $\mathcal{HL}^{\downarrow}$-sentence, $\mathcal{B} = (M, R, V)$ a block tree model of φ and s_1 and s_2 states of $\mathcal{M}$ such that there is a path from s_1 to s_2 but not vice versa. For every formula $\psi \in \mathrm{Sub}(\varphi)$, every state s_3 of $\mathcal{M}$ of the same φ-type as s_2, and every assignment g that maps all free variables in ψ to s_1 or states in M preceding s_1:*

$$\mathcal{B}, g, s_1 \models \psi \quad \Longleftrightarrow \quad \mathcal{B}[u_{s_2}/u_{s_3}], g, s_1 \models \psi.$$

Proof. The proof is by induction on the structure of ψ. Most cases are trivial since s_1 is the only state that has to be considered, e.g., if $\psi = \psi_1 \wedge \psi_2$ we only need to know whether ψ_1 and ψ_2 hold at s_1.

The interesting case is $\psi = \Diamond\xi$. Here, we need to know whether ξ holds at some state s' reachable from s_1. This can only be affected by our substitution if s' is in the replaced subtree, and therefore strictly below s_1. Thus, by our assumption on g, all free variables in ψ are mapped to states different and not reachable from s'. We can conclude

$$\mathcal{B}, g, s' \models \xi \quad \Longleftrightarrow \quad \mathcal{B}, g, s' \models \xi[\mathit{free}/\bot].$$

Hence ξ holds at some state in the replaced subtree rooted at u_{s_2}, if and only if $\xi[\mathit{free}/\bot]$ is in the φ-type of u_{s_2}. The lemma follows because the new subtree has the same φ-type. □

Note that we restricted the choice of g only to those assignments that are relevant when evaluating the sentence φ.

We can use the previous lemma to get some nice restrictions on the block trees under consideration. E.g., we can assume that for every sentence in the type of a node, there is a witness in the node itself or in one of its children.

Lemma 3.7. *Let φ be an $\mathcal{HL}^{\downarrow}$-sentence satisfiable over transitive frames. Then there is a block tree model $\mathcal{B}$ for φ where*

- *every node has at most $|\varphi|$ children,*
- *for every node u with φ-type t and every $\mathcal{HL}^{\downarrow}$-sentence $\psi \in t$, ψ holds at a state in u or at a state in a child of u, and*
- *on every path from the root, infinite or ending at a leaf, every φ-type occurs only once or infinitely often.*

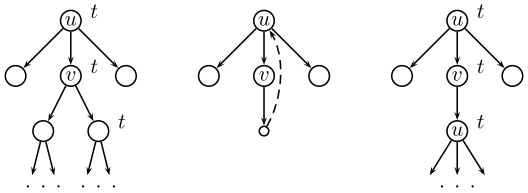

Fig. 3. An infinite block tree, its finite representation, and the infinite block tree obtained from this representation.

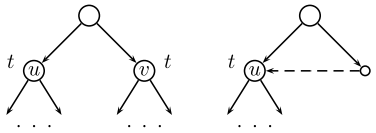

Fig. 4. Elimination of duplicates in the representation of a block tree.

Proof. Let φ be an $\mathcal{HL}^{\downarrow}$-sentence satisfiable over transitive frames and $\mathcal{B}'$ a block tree model for φ.

A block tree satisfying the third condition can be obtained from $\mathcal{B}'$ by applying Lemma 3.6. If there are two nodes u and v on a path, v below u, that have the same φ-type, we can replace the subtree rooted at u by the subtree rooted at v. This technique is standard in tableau decision procedures where blocking ensures termination and possibly infinite satisfying models are constructed from finite tableau branches [12,14,5,20]. In our case, every finite repetition can be reduced to a single occurrence of a φ-type. The resulting structure is still a block tree model for φ.

Now consider some node u and its φ-type t. For every sentence $\psi \in t$, we select some state in the subtree rooted at u such that ψ holds at this state. Let $u_1, \ldots, u_k$ be the corresponding nodes. It is easy to see, following similar ideas as in Lemma 3.6, that the block tree obtained by removing the nodes below u and inserting instead $u_1, \ldots, u_k$ as children of u is again a block tree model of φ. Even more, the type of u is not changed by this replacement.

By successively applying this argument, we obtain a block tree model for φ satisfying the first two conditions. The third one is not affected by this transformation.

Note that some care is needed to make this approach work for infinite models. Basically, we must define a function that assigns to each node of the original model its set of witnesses. The resulting model is obtained by using this function in a straightforward fashion. □

3.3. Deciding $\mathcal{HL}^{\downarrow}$-SAT over transitive frames

We will now finish the proof of Theorem 3.1 by presenting a nondeterministic algorithm that decides $\mathcal{HL}^{\downarrow}$-SAT over transitive frames in exponential time. This algorithm guesses and verifies the finite representation of a block tree model for a given $\mathcal{HL}^{\downarrow}$-sentence φ.

We can transform block trees into finite representations and back in the usual way known from standard tableau algorithms with blocking conditions [12,14,5,20]. This is shown in Fig. 3. Due to Lemma 3.6, the size of the representation can be reduced even further using the ideas underlying the "anywhere blocking" technique, which is used in tableau decision procedures for expressive modal and description logics [30] and related to "global caching" [15,21]. If there are two nodes u and v of the same φ-type which are both the first node of their type on their path from the root, we can replace the subtree rooted at v with the subtree of u. I.e., whenever two nodes have the same φ-type, we can assume that their generated subtrees are equal. We need to check them only once. Hence we can replace every duplicate with a reference, as illustrated in Fig. 4. Therefore, every type occurs at most twice in the representation, and hence the number of nodes is at most exponential.

Such a representation can be described by a structure $(M \uplus C, R, V)$ such that the states in C have no outgoing edges, and a function f from C to M. A state $s \in C$ stands for a repetition of the subtree rooted at $f(s)$, including states from C. As a result, if φ is satisfiable, there is a representation of a block tree model for φ of size at most exponential in the length of φ. Therefore the following decision procedure decides satisfiability nondeterministically in exponential time: guess a representation of a block tree and check whether the model satisfies φ in some state. We call this procedure SATTRANS and make it explicit in Algorithm 1. On input $\varphi \in \mathcal{HL}^{\downarrow}$, it guesses an exponential-size representation $\mathcal{M}, f$ of a block tree and applies a variant MCTRANS of the model checking algorithm MCFULL by Franceschet and de Rijke [16] to check whether φ is satisfiable in the model represented by $\mathcal{M}, f$. In order to handle our finite representations of block trees, SATTRANS guesses the φ-type of all states in C in an initial step, then applies the usual labelling procedure to the states in M via MCTRANS, and finally compares the φ-type of each state s in C with the φ-type of $f(c)$. SATTRANS accepts if and only if the last step was successful and φ occurs in a label of some state in M.

Our variant of MCFULL [16] is called MCTRANS and presented in Fig. 2. This algorithm gets as input a finite representation $\mathcal{M} = (M \uplus C, R, V)$ and f of a block tree, an assignment g and a formula φ and computes a table L of $|\varphi| \times |M \uplus C|$ bits in a combined top-down and bottom-up manner. The bit $L(\psi, m)$ is set if the state m is labelled with ψ, which means

Algorithm 1. SATTRANS(φ)

Input: $\mathcal{HL}^{\downarrow}$-formula φ
Output: accept/reject
Guess exponential-size representation $\mathcal{M} = (M \uplus C, R, V)$, f of a block tree
for all $m \in C$ and *sentences* $\psi \in \mathrm{Sub}(\varphi)$ **do**
  guess $L(\psi, m)$
end for
MCTRANS($\mathcal{M}, g, \varphi$)
for all $m \in C$ **do**
  if $L(m) \neq L(f(m))$ **then reject**
end for
for all $m \in M$ **do**
  if $\varphi \in L(m)$ **then accept**
end for
reject

Algorithm 2. MCTRANS($\mathcal{M}, g, \varphi$)

Input: block tree $\mathcal{M} = (M \uplus C, R, V)$; assignment g; $\mathcal{HL}^{\downarrow}$-formula φ
Output: $L : \mathrm{Sub}(\varphi) \times (M \uplus C) \to \{0, 1\}$
for all $m \in M \uplus C$ and $\psi \in \mathrm{Sub}(\varphi)$ **do**
  $L(\psi, m) \leftarrow 0$
end for
for all $m \in M$ and $p \in \mathrm{PROP} \cup \mathrm{NOM}$ **do**
  if $m \in V(p)$ **then** $L(p, m) \leftarrow 1$
end for
Check($\mathcal{M}, g, \varphi$)

Algorithm 3. Check($\mathcal{M}, g, \varphi$)

if $\varphi = \neg\alpha$ **then** Check$_{\neg}$($\mathcal{M}, g, \alpha$)
if $\varphi = \alpha \wedge \beta$ **then** Check$_{\wedge}$($\mathcal{M}, g, \alpha, \beta$)
if $\varphi = \Diamond\alpha$ **then** Check$_{\Diamond}$($\mathcal{M}, g, \alpha$)
if $\varphi = @_t\alpha$ **then** Check$_{@}$($\mathcal{M}, g, t, \alpha$)
if $\varphi = \downarrow x.\alpha$ **then** Check$_{\downarrow}$($\mathcal{M}, g, x, \alpha$)

Algorithm 4. Check$_{\neg}$($\mathcal{M}, g, \alpha$)

Check($\mathcal{M}, g, \alpha$)
for all $m \in M$ **do**
  if $L(\alpha, m) = 0$ **then** $L(\neg\alpha, m) \leftarrow 1$
end for

Algorithm 5. Check$_{\wedge}$($\mathcal{M}, g, \alpha, \beta$)

Check($\mathcal{M}, g, \alpha$)
Check($\mathcal{M}, g, \beta$)
for all $m \in M$ **do**
  if $L(\alpha, m) = 1$ & $L(\beta, m) = 1$ **then**
    $L(\alpha \wedge \beta, m) \leftarrow 1$
  end if
end for

that the block tree represented by $\mathcal{M}$ and f satisfies ψ at m under g. For our purposes, MCTRANS accepts if $L(\varphi, m) = 1$ for some $m \in M$. For ease of notation, we use $L(\alpha) = \{n \in M \uplus C \mid L(\alpha, n) = 1\}$ and $L(m) = \{\beta \in \mathrm{Sub}(\varphi) \mid L(\beta, m) = 1\}$ for any formula α and state m.

The recursive routine Check called in the last line of MCTRANS is given in Algorithm 3. Depending on the kind of subformula, it calls the subroutines given in Algorithms 4–8.

Theorem 3.8. *SATTRANS* (*Algorithm* 1) *decides* $\mathcal{HL}^{\downarrow}$*-satisiability over transitive frames nondeterministically in exponential time.*

Proof. In the following, we assume that MCTRANS is sound and complete, which can be shown exactly as for MCFULL in [16].

Completeness. If φ is satisfiable, then SATTRANS can guess the finite representation of a block tree model for φ and the φ-types of the states in C. The computation of the φ-types of the states in M works correctly because MCTRANS is complete and we have a witness for every sentence in the type of some node in our representation. The latter follows from Lemma 3.7, which ensures that witnesses are in the node of the state or in one of its children. Therefore, we cut below these witnesses when building the finite representation. Consequently, SATTRANS will accept.

Algorithm 6. Check$_\Diamond(\mathcal{M}, g, \alpha)$

```
Check(M, g, α)
for all m ∈ L(α) do
  for all n with nRm do
    L(◇α, n) ← 1
  end for
end for
```

Algorithm 7. Check$_@(\mathcal{M}, g, t, \alpha)$

```
Check(M, g, α)
{m} ← [V, g](t)
if L(α, m) = 1 then
  for all n ∈ M do
    L(@_t α, n) ← 1
  end for
end if
```

Algorithm 8. Check$_\downarrow(\mathcal{M}, g, x, \alpha)$

```
for all m ∈ M do
  g(x) ← w
  Check(M, g, α)
  if L(α, m) = 1 then L(↓x.α, m) ← 1
  for all ψ ∈ Sub(φ) such that ψ contains x free do
    for all m ∈ M do L(ψ, m) ← 0 end for
  end for
end for
```

Soundness. If SATTRANS accepts, it is straightforward to construct a block tree model from the guessed representation. The only critical point is after the call to MCTRANS, when the φ-types, which have been guessed beforehand, are verified. First, the φ-type of some state $s \in M$ contains only sentences that hold at some state of the represented model below s. This can be ensured by looking only at states in M, not in C. In order for the φ-type of s to contain all sentences that hold below s, only the links represented by states in C should be followed.

Complexity. The first two steps of SATTRANS before the call to MCTRANS can be performed in exponential time because the representation is of at most exponential size. MCTRANS runs in exponential time because of the same argumentation as in Theorem 4.5 of [16]. The time bounds for the remaining steps follow again from the exponential size bound of the representation. □

From Theorem 3.8 and Corollary 3.4 we can conclude Theorem 3.1.

4. Decidability of richer hybrid ↓ logics

This section is concerned with satisfiability over transitive frames, transitive trees, and linear frames for extensions of $\mathcal{HL}^\downarrow$. We extend the logic of the previous section with the @-operator and/or the past modality P and prove undecidability over transitive frames. We will also consider these logics over more restricted frame classes, transitive trees and linear frames, where we will show them to be nonelementarily decidable.

4.1. Transitive frames

Over transitive frames, we cannot retain decidability if we enrich $\mathcal{HL}^\downarrow$ with @ or the backward looking modality P. We prove undecidability in both cases via an undecidable fragment of first-order logic. The notation of such fragments is given in Section 2.

We proceed in two steps. First, we show that [all, (4, 1)]-trans-SAT is undecidable. This is done by a reduction from [all, (0, 1)]-SAT. The undecidability of the latter is a consequence of the undecidability of traditional standard classes that are contained in [all, (0, 1)] [11]. The second step will consist of reductions from [all, (4, 1)]-trans-SAT to $\mathcal{HL}^{\downarrow,@}$-trans-SAT and $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-trans-SAT, respectively. The established lower bounds will already hold for the fragments of the respective hybrid languages consisting of all nominal-free sentences.

Lemma 4.1. [all, (4, 1)]*-trans-SAT is undecidable.*

Proof. In order to obtain a reduction from [all, (0, 1)]-SAT, we will transform a (not necessarily transitive) model satisfying α into a transitive one. Simply taking the transitive closure in most cases adds new pairs to the interpretation of the relation and is not sufficient for keeping the information about which pairs were in the "old" relation and which were not. This problem does not arise if we instead use a variation of the *zig-zag technique* successfully applied in [3] for a reduction

$$
\begin{aligned}
(xRy)^t &= \exists abc\,\big(xRa \wedge bRa \wedge bRc \wedge yRc \\
&\qquad \wedge 0(x) \wedge 1(a) \wedge 2(b) \wedge 3(c) \wedge 0(y)\big),\\
(\neg\alpha)^t &= \neg(\alpha^t),\\
(\alpha \wedge \beta)^t &= \alpha^t \wedge \beta^t,\\
(\exists x\,\alpha)^t &= \exists x\,\big(0(x) \wedge \alpha^t\big).
\end{aligned}
$$

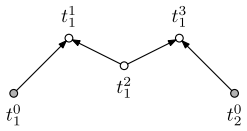

Fig. 5. The translation function and a zig-zag transition.

between a modal and a hybrid language. The core idea of this technique is to simulate an R-step $t_1 R t_2$ in the original model $\mathcal{M} = (D, I)$ by a zig-zag transition in a model $\mathcal{M}' = (D', I')$, where $I'(R)$ is transitive, as shown in Fig. 5.

We define a translation function $(\cdot)^t$ using four extra predicate symbols $0, 1, 2, 3$ according to Fig. 5. The translation of the xRy-atoms reflects the shown zig-zag transition. It is now straightforward to prove the following claim.

Claim. *For each formula α, α is satisfiable iff α^t is satisfiable in some model that interprets R by a transitive relation.*

Proof. We assume that α has no free variables and that each variable is quantified exactly once. This can always be achieved by additional existential quantification and renaming, respectively.

"$\Rightarrow$". Suppose α is satisfied by some model $\mathcal{M} = (D, I)$. We construct a new model $\mathcal{M}^4 = (D^4, I^4)$, where $D^4 = D^0 \cup \cdots \cup D^3$, using $D^i = \{d^i \mid d \in D\}$ for $i = 0, 1, 2, 3$. The interpretation I^4 is defined by

$$
\begin{aligned}
&I^4(R) = \big\{\big(x^0, x^1\big), \big(x^2, x^1\big), \big(x^2, x^3\big), \big(y^0, x^3\big) \mid (x, y) \in I(R)\big\} \quad \text{and}\\
&I^4(P) = D^P, \quad P = 0, 1, 2, 3.
\end{aligned}
$$

$I^4(R)$ codes an $I(R)$-transition from state x to y in $\mathcal{M}$ as a sequence of backward and forward transitions from x_0 to y_0 via x_1, x_2, x_3 as shown in Fig. 5. It is easy to see that $I^4(R)$ is transitive, since there is no state with incoming *and* outgoing $I^4(R)$-edges.

We now show that for all subformulae $\beta(x_1, \ldots, x_m)$ of α and all $d_1, \ldots, d_m \in D'$: $\mathcal{M} \models \beta[d_1, \ldots, d_m]$ iff $\mathcal{M}^4 \models \beta^t[d_1^0, \ldots, d_m^0]$. This immediately implies that $\mathcal{M}^4$ satisfies α^t.

We proceed by induction on β. The base case, $\beta = xRy$, is clear from the construction of $I^4(R)$. The Boolean cases are obvious. For the case $\beta = \exists x\gamma$, we argue

$$
\begin{aligned}
&\mathcal{M} \models \exists x\gamma[d_1, \ldots, d_m]\\
&\quad\iff \exists d \in D\big(\mathcal{M} \models \gamma[d_1, \ldots, d_m, x \mapsto d]\big)\\
&\quad\iff \exists d \in D\big(\mathcal{M}^4 \models \gamma^t\big[d_1^0, \ldots, d_m^0, x \mapsto d^0\big]\big)\\
&\quad\iff \exists d' \in D^4\big(\mathcal{M}^4 \models \big(0(x) \wedge \gamma^t\big)\big[d_1^0, \ldots, d_m^0, x \mapsto d'\big]\big)\\
&\quad\iff \mathcal{M}^4 \models \exists x\big(0(x) \wedge \gamma^t\big)\big[d_1^0, \ldots, d_m^0\big]\\
&\quad\iff \mathcal{M}^4 \models (\exists x\gamma)^t\big[d_1^0, \ldots, d_m^0\big].
\end{aligned}
$$

The second and third line are equivalent due to the induction hypothesis. The equivalence of the third and fourth line follows from the definition of R^4; for the direction from below to above, one must take into account that, because x is interpreted by d' and $0(x)$ is satisfied, d' is indeed some d^0.

"$\Leftarrow$". Let $\mathcal{M} = (D, I)$ be a model satisfying α^t, where $I(R)$ is transitive. We construct a new model $\mathcal{M}' = (D', I')$, where $D' = I(0)$ and

$$
I'(R) = \big\{(d, e) \in \big(D'\big)^2 \mid \exists abc \in D\big((d, a), (b, a), (b, c), (e, c) \in I(R)\ \&\ a \in I(1)\ \&\ b \in I(2)\ \&\ c \in I(3)\big)\big\}.
$$

We now show that for all subformulae $\beta(x_1, \ldots, x_m)$ of α and all $d_1, \ldots, d_m \in D$: $\mathcal{M}' \models \beta[d_1, \ldots, d_m]$ iff $\mathcal{M} \models \beta^t[d_1, \ldots, d_m]$. This immediately implies that $\mathcal{M}'$ satisfies α.

Again, the proof is via induction on β. The base case, $\beta = xRy$, is clear from the construction of $I'(R)$ and the fact that the translation of xRy requires $0(x)$ and $0(y)$. The Boolean cases are obvious. For the case $\beta = \exists x\gamma$, we argue

$$
\begin{aligned}
&\mathcal{M}' \models \exists x\gamma[d_1, \ldots, d_m]\\
&\quad\iff \exists d \in D'\big(\mathcal{M}' \models \gamma[d_1, \ldots, d_m, x \mapsto d]\big)\\
&\quad\iff \exists d \in I(0)\big(\mathcal{M} \models \gamma^t[d_1, \ldots, d_m, x \mapsto d]\big)\\
&\quad\iff \exists d \in D\big(\mathcal{M} \models \big(0(x) \wedge \gamma^t\big)[d_1, \ldots, d_m, x \mapsto d]\big)\\
&\quad\iff \mathcal{M} \models \exists x\big(0(x) \wedge \gamma^t\big)[d_1, \ldots, d_m]\\
&\quad\iff \mathcal{M} \models (\exists x\gamma)^t[d_1, \ldots, d_m].
\end{aligned}
$$

The induction hypothesis is applied between the second and third line. It is obvious that the third line implies the fourth; the backward direction is due to the fact that x is interpreted by d and $0(x)$ is satisfied, hence $d \in I(0)$. □

Since $(\cdot)^t$ is a reduction function, which is even polynomial-time computable, we have established undecidability for [all, (4, 1)]-trans-SAT. □

Theorem 4.2. *$\mathcal{HL}^{\downarrow,@}$-trans-SAT, $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-trans-SAT, and $\mathcal{HL}^{\downarrow,@}_{\mathsf{F},\mathsf{P}}$-trans-SAT are undecidable.*

Proof. We reduce [all, (4, 1)]-trans-SAT to the problems $\mathcal{HL}^{\downarrow,@}$-trans-SAT and $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-trans-SAT, invoking a spy-point argument. A spy-point is a state s of a hybrid model that sees all other states and is named by a fresh nominal i. For details of the spy-point technique see [7,2]. Since our reduction will not make use of any nominals, we can establish this undecidability result for the nominal-free fragments of the hybrid languages in question. We simply treat i as a state variable and bind it to s.

We first consider the case of $\mathcal{HL}^{\downarrow,@}$ and define a translation function $(\cdot)^t$ from [all, (4, 1)] to $\mathcal{HL}^{\downarrow,@}$ by

$$(xRy)^t = @_x \Diamond y, \qquad (\neg\alpha)^t = \neg\big(\alpha^t\big),$$
$$\big(P(x)\big)^t = @_x p, \qquad (\alpha \wedge \beta)^t = \alpha^t \wedge \beta^t,$$
$$(\exists x\alpha)^t = @_i \Diamond \downarrow x.\alpha^t.$$

The (polynomial) reduction function f is defined by

$$f(\alpha) = \downarrow i.\big(\neg\Diamond i \wedge \Diamond\alpha^t\big).$$

In order to show that each formula α is satisfiable iff $f(\alpha)$ is satisfiable, we again assume that α is a sentence. For the "⇒" direction, suppose α is satisfied by a model $\mathcal{M} = (D, I)$. By adding the spy-point s to D, we obtain the hybrid model $\mathcal{M}^h = (M^h, R^h, V^h)$, where $M^h = D \cup \{s\}$, $R^h = I(R) \cup \{(s, d) \mid d \in D\}$, and $V^h(p) = I(P)$. Since I is transitive and the spy-point s cannot be seen by any other state, R^h is transitive, too. Clearly, $\mathcal{M}^h$ satisfies $f(\alpha)$ at s—under any assignment, since $f(\alpha)$ is a sentence.

For the "⇐" direction, suppose $f(\alpha)$ is satisfied at state s of some hybrid model $\mathcal{M} = (M, R, V)$. Then $f(\alpha)$ enforces s to behave as the spy-point. It is easy to see that $\mathcal{M}' = (M - \{s\}, I)$, where $I(R) = R\restriction_{M-\{s\}}$ and $I(P) = V(p)$, satisfies α.

In the case of $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$, we express the @ operator using P, which is possible in the presence of a spy-point and transitivity. We simply re-define $(\cdot)^t$ by

$$(xRy)^t = \mathsf{P}\big(i \wedge \mathsf{F}(x \wedge \mathsf{F}y)\big), \qquad (\neg\alpha)^t = \neg\big(\alpha^t\big),$$
$$\big(P(x)\big)^t = \mathsf{P}\big(i \wedge \mathsf{F}(x \wedge p)\big), \qquad (\alpha \wedge \beta)^t = \alpha^t \wedge \beta^t,$$
$$(\exists x\alpha)^t = \mathsf{P}\big(i \wedge \mathsf{F}\downarrow x.\alpha^t\big).$$

The rest of the proof is the same as for $\mathcal{HL}^{\downarrow,@}$.

The case of $\mathcal{HL}^{\downarrow,@}_{\mathsf{F},\mathsf{P}}$ follows immediately. □

4.2. Transitive trees

Over transitive trees, where decidability of $\mathcal{HL}^{\downarrow}$ is trivial, even the extension $\mathcal{HL}^{\downarrow,@}_{\mathsf{F},\mathsf{P}}$ is decidable. This is an immediate consequence of the decidability of the monadic second-order theory of the countably branching tree, SωS [11]. However, we have to face a nonelementary lower bound in both cases $\mathcal{HL}^{\downarrow,@}$ and $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$. This is obtained by a reduction from the nonelementarily decidable $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-$(\mathbb{N}, >)$-SAT [17]. Here $(\mathbb{N}, >)$ stands for the frame class consisting only of the frame $(\mathbb{N}, >)$.

Theorem 4.3. *$\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-tt-SAT, $\mathcal{HL}^{\downarrow,@}$-tt-SAT, and $\mathcal{HL}^{\downarrow,@}_{\mathsf{F},\mathsf{P}}$-tt-SAT are nonelementarily decidable.*

Proof. Decidability immediately follows from decidability of SωS, using the Standard Translation ST. For the nonelementary lower bound, we reduce $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-$(\mathbb{N}, >)$-SAT to $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-tt-SAT and $\mathcal{HL}^{\downarrow,@}$-tt-SAT, respectively.

Let us first consider $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-tt-SAT. The frame $(\mathbb{N}, >)$ is a special case of a transitive tree. Our language is strong enough to enforce that a transitive tree model is based on $(\mathbb{N}, >)$. We only need to require the following two properties.

(1) Every point has at most one direct successor.
(2) The underlying frame is rooted.

Property (2) is expressed via $\mathsf{PH}\bot$. Property (1) can be formulated in the following way: For any point x, whenever x has some successor, then we name one of the *direct* successors y and ensure that *all* direct successors of x satisfy y. This translates as

$$\lambda = \mathsf{G}\big(\mathsf{F}\top \to \big(\downarrow x.\mathsf{F}^1 \downarrow y.@_x \mathsf{G}^1 y\big)\big),$$

where F^1, P^1, and G^1 refer to direct successors and can be expressed by means of U and S, for example $\mathsf{F}^1\varphi \equiv \mathsf{U}(\varphi, \bot)$. But $\mathsf{U}(\varphi, \psi)$ can be expressed by $\downarrow x.\mathsf{F}(\varphi \wedge \mathsf{H}(\mathsf{P}x \to \psi))$; analogously for $\mathsf{S}(\varphi, \psi)$.

Hence λ is expressible in our language and of constant length. A reduction function f from $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-$(\mathbb{N}, >)$-SAT to $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-tt-SAT is given by $f(\varphi) = \varphi \wedge \lambda \wedge \mathsf{H}\lambda \wedge \mathsf{HG}\lambda \wedge \mathsf{PH}\bot$. It is easy to observe that φ is satisfiable in some linear model (e.g. $(\mathbb{N}, >)$) iff $f(\varphi)$ is satisfiable in some transitive tree.

In the case of $\mathcal{HL}^{\downarrow,@}$-tt-SAT, we rewrite the P operator using a modified spy-point argument. We label one point in the transitive tree by a fresh nominal i and express each occurrence of P in φ using $\downarrow$, a fresh variable v, and i. This is done in the following translation function $(\cdot)^t : \mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}} \to \mathcal{HL}^{\downarrow,@}$.

$$\begin{aligned}
&a^t = a, \quad a \in \mathrm{ATOM}, && (\mathsf{F}\psi)^t = \Diamond\big(\psi^t\big),\\
&(\neg\psi)^t = \neg\big(\psi^t\big), && (\mathsf{P}\psi)^t = \downarrow v.@_i\Diamond\big(\psi^t \wedge \Diamond v\big),\\
&(\psi_1 \wedge \psi_2)^t = \psi_1^t \wedge \psi_2^t, && (\downarrow x.\psi)^t = \downarrow x.\big(\psi^t\big).
\end{aligned}$$

It is easy to see that for each model $\mathcal{M}$ based on $(\mathbb{N}, >)$, for each point $x \in \mathbb{N}$, and for each formula $\varphi \in \mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$, it holds that, whenever i is true at the root 0, then $\mathcal{M}, x \models \varphi \Leftrightarrow \mathcal{M}, x \models \varphi'$.

The point s labelled i represents the root of the frame $(\mathbb{N}, >)$. In the language $\mathcal{HL}^{\downarrow,@}$, it is not possible to express Property (2). This is in fact not necessary if we make sure that we never refer to the past of s in our final translation of φ. Such a "defective" reference can only appear when the @ operator is used in connection with nominals occurring in φ. Let $\mathrm{NOM}(\varphi) = \mathrm{NOM} \cap \mathrm{Sub}(\varphi)$, and let

$$\mu = \bigwedge_{j \in \mathrm{NOM}(\varphi)} @_i \Diamond j.$$

Now the formula $\downarrow i.(\Diamond\varphi^t \wedge \mu)$ does not contain any reference to any point before s. It remains to ensure Property (1). This is done by replacing λ with

$$\lambda' = \Box\big(\Diamond\top \to \downarrow x.\Diamond^1 \downarrow y.@_x \Box^1 y\big)$$

and again expressing $\Diamond^1$ and $\Box^1$ by means of U as above. Now, a reduction function f' from $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-$(\mathbb{N}, >)$-SAT to $\mathcal{HL}^{\downarrow,@}$-tt-SAT is given by $f'(\varphi) = \downarrow i.(\Diamond\varphi^t \wedge \mu \wedge \lambda' \wedge \Box\lambda')$. □

4.3. Linear frames

In the last part of this section we consider linear frames. A frame is called linear if it is irreflexive, transitive, and trichotomous. An important special case is the frame of the natural numbers with the usual ordering relation.

Hybrid $\downarrow$ languages over linear frames have been addressed by Franceschet, de Rijke, and Schlingloff [17]. They showed that satisfiability of $\mathcal{HL}^{\downarrow,@}_{\mathsf{F},\mathsf{P}}$ is nonelementary, even over natural numbers. This result carries over to $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$, because $@_i\varphi$ can be expressed by

$$\mathsf{P}(i \wedge \varphi) \vee (i \wedge \varphi) \vee \mathsf{F}(i \wedge \varphi).$$

While the complexity drops to NP for $\mathcal{HL}^{\downarrow}$, the case of $\mathcal{HL}^{\downarrow,@}$ was left open.

Theorem 4.4. *The satisfiability problem for $\mathcal{HL}^{\downarrow,@}$ over linear frames and over natural numbers is nonelementarily decidable.*

Proof. Only the lower bound needs to be shown. We use a reduction from the satisfiability problem of first order logic over strings, a problem long known to have nonelementary complexity [40].

Strings over a finite alphabet Σ can be represented as

$$\big(\{1, \ldots, n\}, <, (P_\sigma)_{\sigma \in \Sigma}\big),$$

were $<$ is the usual ordering and P_σ a unary relation for every $\sigma \in \Sigma$. As before, these structures can also be used for hybrid reasoning.

We give a translation HT from first order logic into $\mathcal{HL}^{\downarrow,@}$ such that, for every string $\mathcal{S}$ and every first order sentence φ,

$$\mathcal{S} \models \varphi \quad \Longleftrightarrow \quad \mathcal{S}', w_s \models \downarrow s.\big(\mathrm{HT}(\varphi) \wedge \psi\big), \tag{1}$$

where $\mathcal{S}'$ results from $\mathcal{S}$ by adding a spy-point w_s preceding all other states. We use the state variable s to address the spy-point.

$$\mathrm{HT}\big(P_\sigma(x)\big) = @_s\Diamond(x \wedge p_\sigma), \qquad \mathrm{HT}(\neg\varphi) = \neg\mathrm{HT}(\varphi),$$
$$\mathrm{HT}(x = y) = @_s\Diamond(x \wedge y), \qquad \mathrm{HT}(\varphi \wedge \psi) = \mathrm{HT}(\varphi) \wedge \mathrm{HT}(\psi),$$
$$\mathrm{HT}(x < y) = @_s\Diamond(x \wedge \Diamond y), \qquad \mathrm{HT}(\exists x.\varphi) = @_s\Diamond\big(\downarrow x.\mathrm{HT}(\varphi)\big).$$

The "only if" direction of (1) is obvious. For the "if" direction, we have to state that $\mathcal{S}'$ is a string and not just a linear frame. This is done by the formula

$$\psi = \mathit{FL} \wedge \mathit{DISCRETE} \wedge \mathit{UNIQUE}.$$

The precise meaning of ψ is that the subframe generated by w_s, but without the state w_s itself, is a string. This is expressed as follows.

There has to be at least one state in the string and there has to be a start and an end of the string, i.e., a state only preceded by w_s and a state without successor.

$$\mathit{FL} = \big(\Diamond\downarrow x.(@_s\Box\neg\Diamond x)\big) \wedge (\Diamond\Box\bot).$$

The frame is not dense.

$$\mathit{DISCRETE} = \Box\big(\Diamond\top \to (\downarrow x.\Diamond\downarrow y.@_x\Box\Box\neg y)\big).$$

Finally, every state has to carry a unique label from the finite alphabet.

$$\mathit{UNIQUE} = \Box \bigvee_{\sigma\in\Sigma}\Big(\sigma \wedge \bigwedge_{\sigma'\neq\sigma} \neg\sigma'\Big).$$

All other properties of a string are already ensured by linearity. □

5. Hybrid until/since logic over transitive frames and transitive trees

In this section, we will consider $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT and $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-tt-SAT. In [3] it was shown that $\mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$-SAT is EXPTIME-complete.

As for the lower bound, we establish a result as general as possible, namely EXPTIME-hardness of $\mathcal{ML}_{\mathsf{U}}$-trans-SAT and $\mathcal{ML}_{\mathsf{U}}$-tt-SAT.

Theorem 5.1. *$\mathcal{ML}_{\mathsf{U}}$-trans-SAT and $\mathcal{ML}_{\mathsf{U}}$-tt-SAT are EXPTIME-hard.*

Proof. We will reduce the *global* satisfiability problem for $\mathcal{ML}$ to both our (local) problems $\mathcal{ML}_{\mathsf{U}}$-trans-SAT and $\mathcal{ML}_{\mathsf{U}}$-tt-SAT using the same reduction function. The global satisfiability problem is defined by

$$\mathcal{ML}\text{-GLOBSAT} = \{\varphi \in \mathcal{ML} \mid \varphi \text{ is true in } \textit{all} \text{ states of some Kripke model } \mathcal{M}\}.$$

Its EXPTIME-completeness is a direct consequence of the EXPTIME-completeness of $\mathcal{ML}^{\mathsf{E}}$-SAT [38].

It may seem difficult to try reducing this problem over *arbitrary* frames to our satisfiability problem over *transitive* frames. The critical point lies in making a non-transitive model transitive: taking the transitive closure of its relation forces us to add new accessibilities that would disturb satisfaction of $\neg\Diamond$-formulae. Fortunately though, the U operator can make us distinguish the accessibilities in the original model from those that have been added to make the relation transitive. Hence, a translation of $\Diamond\varphi$ should demand: "Make sure that the current state sees a state in which the translation of φ holds and that there is no state in between". This translates as $\mathsf{U}(\varphi, \bot)$ into the modal language.

We define a translation function $(\cdot)^t : \mathcal{ML} \to \mathcal{ML}_{\mathsf{U}}$ by

$$p^t = p, \quad p \in \mathrm{PROP}, \qquad (\varphi \wedge \psi)^t = \varphi^t \wedge \psi^t,$$
$$(\neg\varphi)^t = \neg\big(\varphi^t\big), \qquad (\Diamond\varphi)^t = \mathsf{U}\big(\varphi^t, \bot\big).$$

Using $(\cdot)^t$, we construct a reduction function $f : \mathcal{ML} \to \mathcal{ML}_{\mathsf{U}}$ via $f(\varphi) = \varphi^t \wedge \Box\varphi^t$ (which is clearly computable in polynomial time). It is straightforward to prove the following two claims for each $\varphi \in \mathcal{ML}$.

(1) If $\varphi \in \mathcal{ML}$-GLOBSAT, then $f(\varphi) \in \mathcal{ML}_{\mathsf{U}}$-tt-SAT.
(2) If $f(\varphi) \in \mathcal{ML}_{\mathsf{U}}$-trans-SAT, then $\varphi \in \mathcal{ML}$-GLOBSAT.

Since each transitive tree is a transitive model, (1) and (2) together imply the claim of this theorem.

(1) Suppose φ is satisfied in all states of some Kripke model $\mathcal{M} = (M, R, V)$. By considering the submodel generated by some arbitrary state, we can assume w.l.o.g. that $\mathcal{M}$ has a root w_0.

Due to the *tree model property* (see e.g. [10]), we can assume $\mathcal{M}$ to be tree-like. From $\mathcal{M}$, we construct $\mathcal{M}' = (M, R^+, V)$, which is clearly a transitive tree.

Because of the tree shape of $\mathcal{M}$, we observe that, for each pair $(w, v) \in R$, there exists no $u \in M$ *between* w and v in terms of R^+, i.e. no u such that wR^+u and uR^+v. By means of this observation, we show that, for all states $m \in M$ and all formulae $\psi \in \mathcal{ML}$, it holds that $\mathcal{M}, m \models \psi$ iff $\mathcal{M}', m \models \psi^t$. This claim implies that $\mathcal{M}', w_0 \models \varphi^t \wedge \Box\varphi^t$. We prove it by induction on the structure of ψ. The only interesting case is $\psi = \Diamond\vartheta$, with the following reasoning.

$$\begin{aligned}
\mathcal{M}, m \models \Diamond\vartheta \quad &\Longleftrightarrow \quad \exists n \in M(mRn \,\&\, \mathcal{M}, n \models \vartheta) \\
&\Longleftrightarrow \quad \exists n \in M\big(mRn \,\&\, \mathcal{M}', n \models \vartheta^t\big) \\
&\Longleftrightarrow \quad \exists n \in M\big(mR^+n \,\&\, \mathcal{M}', n \models \vartheta^t \,\&\, \neg\exists u \in M\big(mR^+u \,\&\, uR^+n\big)\big) \\
&\Longleftrightarrow \quad \mathcal{M}', m \models \mathsf{U}\big(\vartheta^t, \bot\big).
\end{aligned}$$

The equivalence of the first and the second line follows from the induction hypothesis. The second and third line are equivalent due to the above observation.

(2) Let $\mathcal{M} = (M, R, V)$ be a transitive model and $w_0 \in M$ such that $\mathcal{M}, w_0 \models f(\varphi)$. Again, we restrict ourselves to the submodel generated by w_0. Hence all states of $\mathcal{M}$ are accessible from w_0.

Define a new Kripke model $\mathcal{M}' = (M, R', V)$ from $\mathcal{M}$, where $R' = \{(w, v) \in R \mid \neg\exists u \in M(wRuRv)\}$. We show that, for all states $m \in M$ and all formulae $\psi \in \mathcal{ML}$, it holds that $\mathcal{M}', m \models \psi$ iff $\mathcal{M}, m \models \psi^t$. Again, we use induction on the structure of ψ with the only interesting case $\psi = \Diamond\vartheta$ and the following argument.

$$\begin{aligned}
\mathcal{M}', m \models \Diamond\vartheta \quad &\Longleftrightarrow \quad \exists n \in M\big(mR'n \,\&\, \mathcal{M}', n \models \vartheta\big) \\
&\Longleftrightarrow \quad \exists n \in M\big((n = w_0 \text{ or } w_0Rn) \,\&\, mRn \,\&\, \neg\exists u(mRuRn) \,\&\, \mathcal{M}, n \models \vartheta^t\big) \\
&\Longleftrightarrow \quad \mathcal{M}, m \models \mathsf{U}\big(\vartheta^t, \bot\big).
\end{aligned}$$

The equivalence of the first and the second line is due to $\mathcal{M}$ being rooted, together with the definition of R' and the induction hypothesis. Now, since $\mathcal{M}, w_0 \models \varphi^t \wedge \Box\varphi^t$, we conclude that for all states $x \in M$, $\mathcal{M}, x \models \varphi^t$. The previous claim implies that $\mathcal{M}'$ globally satisfies φ. □

The upper bounds for $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT and $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-tt-SAT require separate treatment. For $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT, we use an embedding into an appropriate fragment of first-order logic. In order to eliminate transitivity, we "simulate" it syntactically, using the operators U^{++} and S^{++} defined in Section 2.

Lemma 5.2. *For $X \subseteq \{@, \mathsf{E}\}$, the problems $\mathcal{HL}^{X}_{\mathsf{U},\mathsf{S}}$-trans-SAT and $\mathcal{HL}^{X}_{\mathsf{U}^{++},\mathsf{S}^{++}}$-SAT are polynomially reducible to each other.*

Proof. Either problem can be reduced to the other via a simple bijection $f : \mathcal{HL}^{X}_{\mathsf{U},\mathsf{S}} \to \mathcal{HL}^{X}_{\mathsf{U}^{++},\mathsf{S}^{++}}$ or its inverse, respectively. This function simply replaces every occurrence of U or S in the input formula by U^{++} or S^{++}. Obviously, f and f^{-1} can be computed in polynomial time. It is straightforward to inductively verify the following two propositions.

(1) For every $\varphi \in \mathcal{HL}^{X}_{\mathsf{U},\mathsf{S}}$: if φ is satisfied in a state m of some transitive model $\mathcal{M}$, then $\mathcal{M}, m \models f(\varphi)$.
(2) For all $\varphi \in \mathcal{HL}^{X}_{\mathsf{U}^{++},\mathsf{S}^{++}}$: if φ is satisfied in a state m of some model $\mathcal{M} = (M, R, V)$, then the transitive model $\mathcal{M}' = (M, R^+, V)$ satisfies $f^{-1}(\varphi)$ at m. □

Now it is not difficult to obtain a 2EXPTIME upper bound for $\mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$-trans-SAT by an embedding into the loosely μ-guarded fragment μLGF of first-order logic whose satisfiability problem is 2EXPTIME-complete [23]. Only the E operator requires a more careful analysis.

Theorem 5.3. *$\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT is in 2EXPTIME.*

Proof. We first embed $\mathcal{HL}^{@}_{\mathsf{U}^{++},\mathsf{S}^{++}}$ into the loosely μ-guarded fragment μLGF of first-order logic [23]. Since the satisfiability problem for μLGF-sentences is 2EXPTIME-complete [23], we obtain a 2EXPTIME upper bound for $\mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$-trans-SAT by Lemma 5.2. As a second step, we will show a reduction from $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT to $\mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$-trans-SAT.

For the embedding into μLGF, we enhance the Standard Translation ST (see Section 2) by the rule

$$\mathrm{ST}_x\big(\mathsf{U}^{++}(\varphi, \psi)\big) = \exists y\big[xR^+y \wedge \mathrm{ST}_y(\varphi) \wedge \forall z\big(\big(xR^+z \wedge zR^+y\big) \to \mathrm{ST}_z(\psi)\big)\big]$$

and an analogous rule for $\mathrm{ST}_x(\mathsf{S}^{++}(\varphi,\psi))$. ST_y and ST_z are defined by exchanging x, y, z cyclically. The xR^+y atoms can be expressed by

$$\big[\mathrm{LFP}W(x,y).\big(xRy \vee \exists z(zRy \wedge xWz)\big)\big]xy,$$

yielding a μLGF-sentence with three variables. (If U^{++} operators are nested, variables can be "recycled".) The constants from the translations of nominals can be eliminated introducing new variables as shown in [22]. The whole translation only requires time polynomial in the length of the input formula.

The more expressive language $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$ can be embedded into $\mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$ using a spy-point argument and exploiting the fact that we are restricted to transitive frames. Due to transitivity, $\mathsf{E}\varphi$ can be expressed by $@_i\Diamond\varphi$ if i is the name of a spy-point. Hence, if we take the translation $(\cdot)^t : \mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}} \to \mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$ that simply replaces all occurrences of E as shown, we obtain a reduction function $f : \mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT $\to \mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$-trans-SAT by setting $f(\varphi) = i \wedge \neg\Diamond i \wedge \Diamond\varphi^t$.

Clearly, f is computable in polynomial time. It is straightforward to show that φ is satisfiable if and only if $f(\varphi)$ is: if $\varphi \in \mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$ is satisfied at some point of some transitive model, add the spy-point s and the corresponding accessibilities. The new model satisfies $f(\varphi)$ at s. For the converse, if a transitive model satisfies $f(\varphi)$ at some point s, then s must be a spy-point, and φ^t is satisfied at another point m. After removing s and the corresponding accessibilities, the remaining model satisfies φ at m. □

A note on the discrepancy between the upper and lower bound for $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT. Since the 2EXPTIME result for μLGF in [23] holds for sentences without constants only, constants—which arise from the translation of nominals—must be reformulated using new variables. This causes an unbounded number of variables in the first-order vocabulary, because we have no restriction on the number of nominals in our hybrid language.

Could we assume that the number of nominals were bounded, then the described reduction would yield guarded fixpoint sentences of bounded width. In this case, satisfiability is EXPTIME-complete [23]. It is not known whether, in the case of a bounded number of variables, but an arbitrary number of constants, satisfiability for μLGF-sentences also decreases from 2EXPTIME to EXPTIME, as is the case for the fragment without the μ operator [42]. If there were a positive answer to this question, an EXPTIME upper bound for our satisfiability problem would follow.

We will now show that $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-tt-SAT is in EXPTIME, using an embedding into $\mathcal{PDL}_{\mathrm{tree}}$, the propositional dynamic logic for sibling-ordered trees [27,28]. Finite, node-labelled, sibling-ordered trees are the logical abstraction of XML (eXtensible Markup Language) documents. In [1], it was shown that satisfiability of $\mathcal{PDL}_{\mathrm{tree}}$ formulae at the root of finite trees ($\mathcal{PDL}_{\mathrm{tree}}$-SAT) is decidable in EXPTIME.

Since we intend to give an embedding into $\mathcal{PDL}_{\mathrm{tree}}$, we first introduce its syntax and semantics. $\mathcal{PDL}_{\mathrm{tree}}$ is the language of propositional dynamic logic with four atomic programs `left`, `right`, `up`, and `down` that are associated with the relations "left sister", "right sister", "parent", and "daughter" in trees. It consists of all formulae of the form

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \langle\pi\rangle\varphi,$$

where $p \in \mathrm{ATOM}$ and π is a program. Programs are defined by

$$\pi ::= \mathtt{left} \mid \mathtt{right} \mid \mathtt{up} \mid \mathtt{down} \mid \pi;\pi' \mid \pi \cup \pi' \mid \pi^* \mid \varphi?,$$

where φ is a formula. We abbreviate $[\pi]\varphi := \neg\langle\pi\rangle\neg\varphi$ and $a^+ := a; a^*$ for atomic programs a.

A $\mathcal{PDL}_{\mathrm{tree}}$ model is a multi-modal model $\mathcal{M} = (T, R_{\mathtt{down}}, R_{\mathtt{right}}, V)$, where T is a finite tree with an order relation on all immediate successors of any node, $R_{\mathtt{down}}$ is the successor relation and $R_{\mathtt{right}}$ is the "next-sister" relation. The set of relations is extended to arbitrary programs as follows:

$$R_{\mathtt{up}} = R^-_{\mathtt{down}}, \qquad R_{\pi\cup\pi'} = R_\pi \cup R_{\pi'},$$
$$R_{\mathtt{left}} = R^-_{\mathtt{right}}, \qquad R_{\pi^*} = R^*_\pi,$$
$$R_{\pi;\pi'} = R_\pi \circ R_{\pi'}, \qquad R_{\varphi?} = \big\{(m,m) \mid \mathcal{M}, m \models \varphi\big\}.$$

The satisfaction relation for atomic formulae and Booleans is defined as for hybrid logic. The modal case is given by

$$\mathcal{M}, m \models \langle\pi\rangle\varphi \quad \text{iff} \quad \exists n \in T(mR_\pi n \,\&\, \mathcal{M}, n \models \varphi).$$

A formula φ is *satisfiable* iff there exists a model $\mathcal{M} = (T, R_{\mathtt{down}}, R_{\mathtt{right}}, V)$ such that $\mathcal{M}, m \models \varphi$, where m is the root of T.

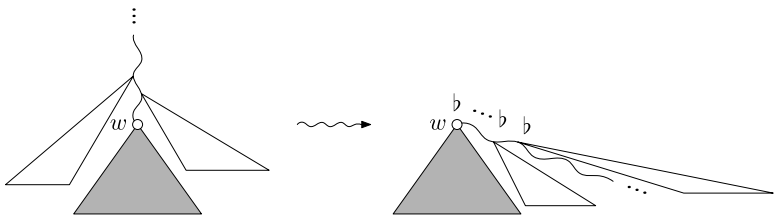

Fig. 6. Making predecessors successors.

Theorem 5.4. *$\mathcal{HL}^{\mathsf{E}}_{\mathsf{U,S}}$-tt-SAT is in* EXPTIME.

Proof. We reduce $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U,S}}$-trans-SAT to $\mathcal{PDL}_{\text{tree}}$-SAT and define a translation $(\cdot)^t : \mathcal{HL}^{\mathsf{E}}_{\mathsf{U,S}} \to \mathcal{PDL}_{\text{tree}}$ by

$$
\begin{aligned}
&p^t = p, \quad p \in \text{ATOM}, && (\mathsf{E}\varphi)^t = \langle \mathtt{up}^*; \mathtt{down}^* \rangle \varphi^t,\\
&(\neg\varphi)^t = \neg(\varphi^t), && \big(\mathsf{U}(\varphi,\psi)\big)^t = \langle (\mathtt{down}; \psi^t?)^*; \mathtt{down} \rangle \varphi^t,\\
&(\varphi \wedge \psi)^t = \varphi^t \wedge \psi^t, && \big(\mathsf{S}(\varphi,\psi)\big)^t = \langle (\mathtt{up}; \psi^t?)^*; \mathtt{up} \rangle \varphi^t.
\end{aligned}
$$

Nominals are translated into atomic propositions, and we need to enforce that the translation of each nominal is true at exactly one point by requiring

$$
\begin{aligned}
\nu(i) = \langle \mathtt{down}^* \rangle i \wedge [\mathtt{down}^*] \big(i \to \big(&[\mathtt{down}^+] \neg i \wedge [\mathtt{up}^+] \neg i\\
&\wedge [\mathtt{up}^*; \mathtt{left}^+; \mathtt{down}^*] \neg i \wedge [\mathtt{up}^*; \mathtt{right}^+; \mathtt{down}^*] \neg i \big)\big)
\end{aligned}
$$

for each nominal i. As a reduction function, we have

$$
f(\varphi) = \langle \mathtt{down}^* \rangle \varphi^t \wedge \bigwedge_{i \in N(\varphi)} \nu(i),
$$

where $N(\varphi)$ is the set of all nominals occurring in the hybrid formula φ.

It is clear that f is computable in polynomial time and straightforward to show that φ is satisfiable if and only if $f(\varphi)$ is. Suppose φ is satisfiable in some finite transitive tree model $\mathcal{M} = (M, R, V)$ based on the tree (M, R') with root w. Then $f(\varphi)$ is satisfiable in w of the $\mathcal{PDL}_{\text{tree}}$ model based on the tree (M, R'), equipped with the valuation V. For the converse, if $f(\varphi)$ is satisfied at the root of some $\mathcal{PDL}_{\text{tree}}$ model $\mathcal{M} = (M, R_{\mathtt{down}}, R_{\mathtt{right}}, V)$, then φ^t is true at some point w, and each nominal is true at exactly one point of $\mathcal{M}$. Hence $(M, R^+_{\mathtt{down}}, V)$, where $R^+_{\mathtt{down}}$ is the transitive closure of $R_{\mathtt{down}}$, is a hybrid transitive tree model satisfying φ at w.

Now there is one drawback in the reduction via f. According to our definition of a tree, it is not necessary that a (transitive) tree is finite or has a root. A node can have infinitely many successors, or there may be an infinitely long forward or backward path from some point. For most practical applications these cases are certainly hardly of interest, but we still want t include them for generality. If we do allow for infinite depth or width, the above translation into $\mathcal{PDL}_{\text{tree}}$—which is interpreted over finite, rooted trees—is not sufficient.

To overcome this problem, it suffices to re-examine the proof for the EXPTIME upper bound of $\mathcal{PDL}_{\text{tree}}$-satisfiability in [1]. This proof in fact covers a more general result, namely that satisfiability of $\mathcal{PDL}_{\text{tree}}$ formulae over not necessarily finite trees is in EXPTIME.

Since "our" tree models do not need to have roots, we first observe that satisfiability over *rooted* transitive trees is reducible to satisfiability over arbitrary transitive trees: having a root is expressible by $\mathsf{PH}\bot$ in our language. Since the lower bound from Theorem 5.1 holds with respect to rooted transitive trees, it also holds for arbitrary ones.

In order to obtain the upper bound with respect to arbitrary transitive trees, modify the above reduction. The basic idea is to turn the backward path from the node w (which is to satisfy φ) into a forward path, such that w becomes the root of the transformed model. Thus all predecessors of w (and *their* predecessors) become successors and must be marked by a fresh proposition $\flat$. (See Fig. 6.)

We first construct a new translation $(\cdot)^{t\flat}$ from $(\cdot)^t$ retaining all but the U/S-cases. For U/S, we replace all occurrences of the programs $\mathtt{down}$ and $\mathtt{up}$ by programs that incorporate the new structure and the fact that for $\flat$-nodes, their predecessors used to be their successors, and their $\flat$-successors used to be their predecessors. We define

$$
\big(\mathsf{U}(\varphi,\psi)\big)^{t\flat} = \langle (\mathtt{dn}'; \psi^{t\flat}?)^*; \mathtt{dn}' \rangle \varphi^{t\flat} \quad \text{and} \quad \big(\mathsf{S}(\varphi,\psi)\big)^{t\flat} = \langle (\mathtt{up}'; \psi^{t\flat}?)^*; \mathtt{up}' \rangle \varphi^{t\flat},
$$

where

$$
\mathtt{dn}' = (\mathtt{down}; \neg\flat?) \cup (\flat?; \mathtt{up}) \quad \text{and} \quad \mathtt{up}' = (\neg\flat?; \mathtt{up}) \cup (\flat?; \mathtt{down}; \flat?).
$$

Note that we do not change the translation of $\mathsf{E}\varphi$. It now remains to enforce that there is exactly one path at whose every node $\flat$ is true. This means that $\flat$ must be true at the root node and at exactly one successor of each node satisfying $\flat$. It can be expressed by

$$\beta = \flat \wedge \big[\mathtt{down}^*\big]\big(\flat \to \big(\big[\mathtt{left}^+\big]\neg\flat \wedge \big[\mathtt{right}^+\big]\neg\flat \wedge \langle\mathtt{down}\rangle\flat\big)\big) \wedge \big[\mathtt{down}^*\big]\big(\neg\flat \to [\mathtt{down}]\neg\flat\big).$$

It is now straightforward to show that $f^\flat$, given by $f^\flat(\varphi) = \varphi^{t\flat} \wedge \beta \wedge \bigwedge_{i\in N(\varphi)} \nu(i)$, is indeed a reduction function. Note that $\varphi^{t\flat}$ replaces $\langle\mathtt{down}^*\rangle\varphi^{t\flat}$ because we have turned w into the new root node. □

6. Conclusion

We have established two groups of complexity results for hybrid logics over three temporally relevant frame classes: transitive frames, transitive trees, and linear frames.

First, we have "tamed" $\mathcal{HL}^{\downarrow}$ over transitive frames showing that $\mathcal{HL}^{\downarrow}$-trans-SAT is NEXPTIME-complete. The key step of our proof was to find a finite representation of transitive models for this logic. In contrast, we proved that $\mathcal{HL}^{\downarrow,@}$-trans-SAT and $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-trans-SAT are undecidable. We also showed in [32] that the *multi-modal* variant of $\mathcal{HL}^{\downarrow}$ over transitive frames and frames with equivalence relations is undecidable.

Over transitive trees, we showed that three enrichments of $\mathcal{HL}^{\downarrow}$ are decidable, albeit nonelementarily, namely $\mathcal{HL}^{\downarrow,@}$-tt-SAT, $\mathcal{HL}^{\downarrow}_{\mathsf{F},\mathsf{P}}$-tt-SAT, and $\mathcal{HL}^{\downarrow,@}_{\mathsf{F},\mathsf{P}}$-tt-SAT. Concerning linear frames, we obtained the same result for $\mathcal{HL}^{\downarrow,@}$-lin-SAT, an issue left open in [17].

In the third part of this article, we established an EXPTIME lower bound for $\mathcal{ML}_{\mathsf{U}}$-trans-SAT and $\mathcal{ML}_{\mathsf{U}}$-tt-SAT and matched the latter with an EXPTIME upper bound for $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-tt-SAT. This is the same complexity as for satisfiability over *arbitrary* frames for the same language. As for $\mathcal{HL}^{\mathsf{E}}_{\mathsf{U},\mathsf{S}}$-trans-SAT, we have given a 2EXPTIME upper bound. We conjecture EXPTIME-completeness.

Over linear frames, the complexity of hybrid U/S logic is still open. As a special case, satisfiability of $\mathcal{HL}^{@}_{\mathsf{U},\mathsf{S}}$ over $(\mathbb{N}, >)$ and of $\mathcal{ML}_{\mathsf{U}}$ over linear orders is known to be PSPACE-complete [17,36].

Acknowledgements

We thank Balder ten Cate, Massimo Franceschet, Maarten Marx, and an unknown referee for helpful suggestions, discussions, and comments.

References

[1] Loredana Afanasiev, Patrick Blackburn, Ioanna Dimitriou, Bertrand Gaiffe, Evan Goris, Maarten Marx, Maarten de Rijke, PDL for ordered trees, Journal of Applied Non-Classical Logics 15 (2) (2005) 115–135.
[2] Carlos Areces, Patrick Blackburn, Maarten Marx, A road-map on complexity for hybrid logics, in: Proc. of the 13th CSL, 1999, in: Lecture Notes in Computer Science, vol. 1683, Springer, 1999, pp. 307–321.
[3] Carlos Areces, Patrick Blackburn, Maarten Marx, The computational complexity of hybrid temporal logics, Logic Journal of the IGPL 8 (5) (2000) 653–679.
[4] Carlos Areces, Patrick Blackburn, Maarten Marx, Hybrid logics: Characterization, interpolation and complexity, Journal of Symbolic Logic 66 (3) (2001) 977–1010.
[5] Franz Baader, Martin Buchheit, Bernhard Hollunder, Cardinality restrictions on concepts, Artificial Intelligence 88 (1–2) (1996) 195–213.
[6] Patrick Blackburn, Representation, reasoning, and relational structures: a hybrid logic manifesto, Logic Journal of the IGPL 8 (3) (2000) 339–365.
[7] Patrick Blackburn, Jerry Seligman, Hybrid languages, Journal of Logic, Language and Information 4 (3) (1995) 251–272.
[8] Patrick Blackburn, Jerry Seligman, What are hybrid languages? in: M. Kracht, M. de Rijke, H. Wansing, M. Zakharyaschev (Eds.), Advances in Modal Logic, in: CSLI Publications, vol. 1, Stanford University, 1998, pp. 41–62.
[9] Patrick Blackburn, Miroslava Tzakova, Hybrid languages and temporal logic, Logic Journal of the IGPL 7 (1) (1999) 27–54.
[10] Patrick Blackburn, Maarten de Rijke, Yde Venema, Modal Logic, Cambridge Tracts in Theoretical Computer Science, vol. 53, Cambridge University Press, 2001.
[11] Egon Börger, Erich Grädel, Yuri Gurevich, The classical decision problem, in: Perspectives of Mathematical Logic, Springer, 1997.
[12] Martin Buchheit, Francesco M. Donini, Andrea Schaerf, Decidable reasoning in terminological knowledge representation systems, Journal of Artificial Intelligence Research 1 (1993) 109–138.
[13] Anuj Dawar, Martin Otto, Modal characterisation theorems over special classes of frames, in: Proc. of the 20th LICS, IEEE Computer Society, 2005, pp. 21–30.
[14] Stéphane Demri, Uniform and non uniform strategies for tableaux calculi for modal logics, Journal of Applied Non-Classical Logics 5 (1) (1995).
[15] Francesco M. Donini, Fabio Massacci, EXPTIME tableaux for ALC, Artificial Intelligence 124 (2000) 87–138.
[16] Massimo Franceschet, Maarten de Rijke, Model checking hybrid logics (with an application to semistructured data), Journal of Applied Logic 4 (3) (2006) 279–304.
[17] Massimo Franceschet, Maarten de Rijke, Bernd-Holger Schlingloff, Hybrid logics on linear structures: Expressivity and complexity, in: Proc. of the 10th TIME, IEEE Computer Society, 2003, pp. 166–173.
[18] Harald Ganzinger, Christoph Meyer, Margus Veanes, The two-variable guarded fragment with transitive relations, in: Proc. of the 14th LICS, IEEE Computer Society, 1999, pp. 24–34.
[19] Valentin Goranko, Hierarchies of modal and temporal logics with reference pointers, Journal of Logic, Language and Information 5 (1) (1996) 1–24.

[20] Rajeev Goré, Tableau methods for substructural logics, in: M. D'Agostino, D.M. Gabbay, R. Hähnle, J. Posegga (Eds.), Handbook of Tableau Methods, Springer, 1999.
[21] Rajeev Goré, Linh Anh Nguyen, EXPTIME tableaux with global caching for description logics with transitive roles, inverse roles and role hierarchies, in: Proc. of the 16th TABLEAUX, in: Lecture Notes in Computer Science, vol. 4548, Springer, 2007, pp. 133–148.
[22] Erich Grädel, On the restraining power of guards, Journal of Symbolic Logic 64 (1999) 1719–1742.
[23] Erich Grädel, Igor Walukiewicz, Guarded fixed point logic, in: Proc. of the 14th LICS, IEEE Computer Society, 1999, pp. 45–54.
[24] Neil Immerman, Alexander M. Rabinovich, Thomas W. Reps, Shmuel Sagiv, Greta Yorsh, The boundary between decidability and undecidability for transitive-closure logics, in: Proc. of the 13th CSL, 2004, in: Lecture Notes in Computer Science, vol. 3210, Springer, 2004, pp. 160–174.
[25] Emanuel Kieronski, EXPSPACE-complete variant of guarded fragment with transitivity, in: Proc. of the 19th STACS, 2002, in: Lecture Notes in Computer Science, vol. 2285, Springer, 2002, pp. 608–619.
[26] Emanuel Kieronski, The two-variable guarded fragment with transitive guards is 2EXP-TIME-hard, in: Proc. of the 6th FoSSaCS, 2003, in: Lecture Notes in Computer Science, vol. 2620, Springer, 2003, pp. 299–312.
[27] Marcus Kracht, Syntactic codes and grammar refinement, Journal of Logic, Language and Information 4 (1) (1995) 41–60.
[28] Marcus Kracht, Inessential features, in: Selected Papers of 1st LACL, 1996, in: Lecture Notes in Computer Science, vol. 1328, Springer, 1997, pp. 43–62.
[29] Richard E. Ladner, The computational complexity of provability in systems of modal propositional logic, SIAM Journal on Computing 6 (3) (1977) 467–480.
[30] Carsten Lutz, Carlos Areces, Ian Horrocks, Ulrike Sattler, Keys, nominals, and concrete domains, Journal of Artificial Intelligence Research 23 (2005) 667–726.
[31] Maarten Marx, Narcissists, stepmothers and spies, in: Proc. of Intl. Workshop on Description Logic, vol. 53, CEUR, 2002.
[32] Martin Mundhenk, Thomas Schneider, Undecidability of multi-modal hybrid logics, Electronic Notes in Theoretical Computer Science 174 (6) (2007) 29–43.
[33] Hiroakira Ono, Akira Nakamura, On the size of refutation Kripke models for some linear modal and tense logics, Studia Logica 34 (1980) 325–333.
[34] Christos H. Papadimitriou, Computational Complexity, Addison–Wesley, 1994.
[35] Arthur Prior, Past, Present and Future, Clarendon Press, Oxford, 1967.
[36] Mark Reynolds, The complexity of the temporal logic with "until" over general linear time, Journal of Computer and System Sciences 66 (2) (2003) 393–426.
[37] A. Prasad Sistla, Edmund M. Clarke, The complexity of propositional linear temporal logics, Journal of the ACM 32 (3) (1985) 733–749.
[38] Edith Spaan, Complexity of modal logics, PhD thesis, ILLC, University of Amsterdam, 1993.
[39] Edith Spaan, The complexity of propositional tense logics, in: Maarten de Rijke (Ed.), Diamonds and Defaults, Kluwer, 1993, pp. 287–307.
[40] Larry Stockmeyer, The complexity of decision problems in automata and logic, PhD thesis, MIT, 1974.
[41] Wieslaw Szwast, Lidia Tendera, On the decision problem for the guarded fragment with transitivity, in: Proc. of the 16th LICS, IEEE Computer Society, 2001, pp. 147–156.
[42] Balder ten Cate, Massimo Franceschet, Guarded fragments with constants, Journal of Logic, Language and Information 14 (3) (2005) 281–288.
[43] Balder ten Cate, Massimo Franceschet, On the complexity of hybrid logics with binders, in: Proc. of the 19th CSL, in: Lecture Notes in Computer Science, vol. 3634, Springer, 2005, pp. 339–354.